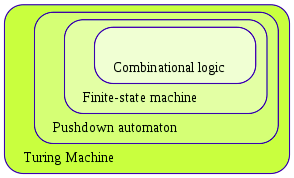
# Assignment 4 – Finite State Machines

## What is a finite state machine?

(Ref 1) A finite state machine is a machine that can calculate exactly one output (state) of an finite amount of states. The finite state machine (also known as an FSMs or Finite state automata) can change its output depending on its inputs. This change from an old output to a new output given a change in an input is called a transition. There are two types of finite state machines, deterministic (synchronous) finite state machines and non-deterministic (asynchronous) finite state machines. The idea of the two types are very similar but there are small differences. A nondeterministic finite machine can be in several states at once, whereas this is impossible for a deterministic finite machine.

Finite state machines are very simplistic. They remember their current state as well as all other states possible for that FSM. It does not store previous states. They also only have a finite amount of memory generally. Alan Turing was the first person to develop an new class of automata (as see in figure 1) that had an infinite amount of memory. It is a good example of what a CPU is in modern society. The only thing that held the Turing Machine back was how fast it could move it’s read head around it’s storage medium, which were holes punched in tape. These holes were called cells and were in a one-dimensional array. Alan Turing’s automata then went on to help solve the enigma cipher.

  
Figure 1

FSM’s play a huge role in our society, they run our vending machines by dispensing products when the proper change is entered, elevators by determining the fastest route depending on where riders are going and traffic lights by changing colours depending on how long cars are waiting and how long they have been waiting. FSM’s are reliable and that is why there are used everywhere.

### What’s the history of FSM’s?

To be able to talk about the history of FSM’s we first have to talk about what automata theory is. The word automata stems from the word automaton which means ‘a self operating machine’ or a machine that is design to automatically follow given instructions. The end of the 19th, start of the 20th century marked the beginning of a new revolution in automata, also know as “The Golden Age of Automata” (Ref 2). During this time automata theory was developed. This theory began to be practiced by Paris clock workers.

(Ref 3) Automata theory is still being practiced to this day in the design of new processor architectures. The most advanced automata proposed is called AREE, a wind-powered rower designed for the harsh climate on Venus by NASA.

Finite State Machines come from a branch of Computer Science, automata theory, which we just talked about.

The idea of the finite state machine was developed in the early 40’s. Doctors, scientists and biologists all had an interest in FSM’s but it was mostly developed by two neurophysiologists Walter Pitts and Warren McCulloch and were the first to publish a research paper, named “A logical calculus immanent in nervous activity”. This paper is what pushed forward the theory of automata to what it is today.

### How do FSM’s work?

Finite state automata are devices, whether that be a software or hardware device that responds to inputs (external events) and produces outputs (actions). The outputs of the machine depend on its current state e.g. the past history of the machine (Ref 4).

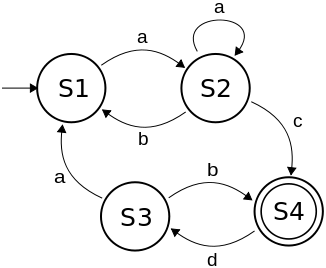
Inputs could be such things like a switch on a computer being pressed, which created an output of the computer turning on or off. Though there is only one switch, there are two events. This means that the output depends on the current state of the switch (system). This implies that the computer has two states, on or off. The number of states is a finite number. This is where the term finite comes from in finite state machine.

Figure 2:2

Shown in the example above (figure 1:2) is a finite state machine. In the FSM we have multiple states: S1, S2, S3 and S4. All taking are either taking an input a, b, c, d or no input at all, which will end in the state not changing. All automata work this way, they can take multiple inputs but can only output one output which changes the state. Below I have generated a truth table that I will use to describe what the finite state machine is doing.

Example inputs to this FSM could be:

- acda

- aabacdbda

- abacda

An example of an input string that wouldn’t work would be:

- baabacda (As S1 cannot take an input b at the starting point)

- aacbda (As there is no option for a transition from c to b at S4)

State table for figure 1:2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Input A | Input B | Input C | Input D | Current State | Next State | Output |
| 0 | 0 | X | X | S1 | S1 | NULL |
| 1 | 0 | X | X | S1 | S2 | NULL |
| 0 | 0 | 0 | X | S2 | S2 | NULL |
| 1 | 0 | 0 | X | S2 | S2 | NULL |
| 0 | 1 | 0 | X | S2 | S1 | NULL |
| 0 | 0 | 1 | X | S2 | S4 | NULL |
| X | X | X | 1 | S4 | S3 | NULL |
| X | X | X | 0 | S4 | S4 | NULL |
| 0 | 0 | X | X | S3 | S3 | NULL |
| 0 | 1 | X | X | S3 | S4 | NULL |
| 1 | 0 | X | X | S3 | S1 | NULL |

In this example I have not stated what the output of each state is, but for example: the output of S1 could be to turn on the element in a kettle after the switch (input a) is pressed; then S2 could be to not let as much electricity through the element depending a thermometer (input b); once S3 receives a signal from a thermometer, it could turn off the kettle; S4 have an input be a sensor that detects if the kettle is sitting on the stand and sends a signal back to S4 to make sure the kettle is off unless a button that sets the temperature you want to water to be brings it back to S1 to turn on the kettle.

### What can FSM’s be used for?

Example 1: Traffic Light (Ref 5)

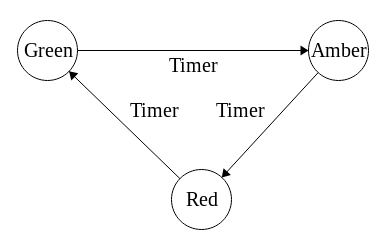


Figure 3

States: Red, Amber, Green

Transitions:

- The only inputs that this FSM relies on are: A timer and the past state as shown in figure 3

State 1: Green can only be activated if the past state is red.  
State 2: Amber can only be activated if the past state is green.  
State 3: Red can be a start state or can be activated if the past state is amber.

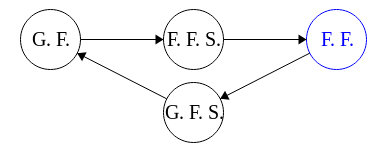
Example 2: Elevator (Ref 6)

Figure 4

G.F. - ground Floor and G.F.S – ground floor switch

F.F – First Floor and F.F.S – first floor switch

States: Ground floor, ground floor slow down switch, first floor slow down switch and first floor.

Transitions:

- The elevator takes an input from a panel where riders decide where they want to go, then other inputs from limit switches on the elevator tracks so it knows when to slow down.

State 1: Ground floor state is our start state.  
State 2: Ground floor slow down switch is ignored if past state is ground floor.  
State 3: First floor slow down switch is activated if past state is ground floor.  
State 4: First floor state is activated if past state is first floor slow down switch

Example 3: Gear Shifter

States: Reverse, Neutral, Gear 1, Gear 2

Transitions:

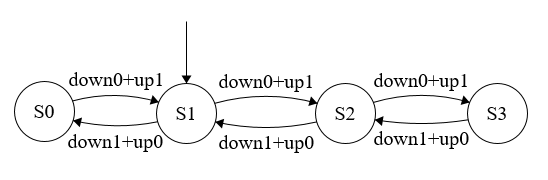
* FSM takes inputs from two buttons on the steering wheel. When the first button is pressed it will go up a state, and when the second button is pressed it will go down a gear. We will refer to these buttons as up and down respectively.

State 0: Reverse can only be activated if past state is neutral.

State 1: Neutral gear is our start state.

State 2: Gear 1 can only be activated if past gear is neutral.

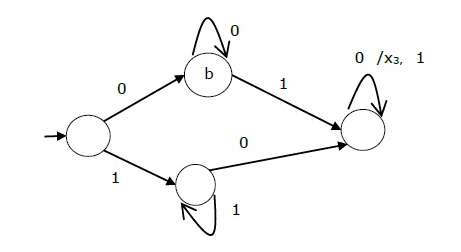
State 3: Gear 2 can only be activated if past gear is gear 1



# Mealy and Moore Machines

### What is a Mealy FSM?

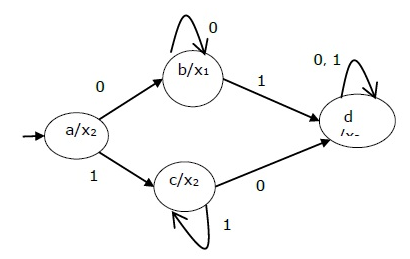
George H. Mealy proposed the idea of a machine in a 1955 paper called “A Method for synthesizing sequential circuits”. They after named the machine the ‘Mealy Machine’. This FSM depends on its present state and input, “it associates its output with the transitions” (Ref 7, Slide 17).



Figure

Shown in figure 5 is an example Mealy FSM. As can be seen the outputs are happening on the transition as opposed to happening on the state.

What is a Moore FSM?  
Edward F. Moore proposed the idea of a machine in a 1956 paper called “Gedanken-experiments on Sequential Machines”. They after named the machine the ‘More Machine’. This FSM associates its state with its output. (Ref 7, Slide 17).



Figure

Shown in figure 6 is an example Moore FSM. As can be seen the outputs are happening on the state.

Advantages and Disadvantages of Mealy and Moore  
Both FSMs have their advantages. (Ref 8) Mealy FSMs can be seen as more complex than Moore FSMs as imagining the output happening on the transition of states can seem confusing, but in general Mealy has less states than Moore. Moore has usually more of a delay when computing, outputs are generally outputs will be computed on the next clock cycle rather than Mealy which reacts within the same clock cycle; this is because the Mealy FSM is able to react as soon as it activates the transition, the Moore FSM has to wait until after the transition. (Ref 9) Mealy is asynchronous and Moore is synchronous. Mealy FSMs are harder to design.

## Full FSM Example

I came up with this example not by looking on the internet. I went around my house looking for simplistic electronics. I first looked at the secondary controls of a car e.g. the indicator level and the buttons on the steering wheel. But I soon decided that they had too many states.

I found a hair dryer. I looked at all the possible states; decided to use the Moore FSM as it was more simplistic than a mealy FSM but aside from that, Moore suited my FSM better as I wanted the outputs of the FSM to be determined by its present state. As shown in the diagram below.

## 

F0+H1+B0

F0+H1+B1

F0+H1+B1

F0+H1+B0

F1+H1+B1

F1+H1+B0

F0+H1+B1

F0+H1+B0

B0

B0

B0

B0

F1+H1+B1

F1+H1+B0

F0+H1+B1

F0+H1+B0

F1+H1+B0

F1+H1+B1

In my finite state machine, there are three inputs. A button that manipulates the fan, the heat of the air and a button that decides the polarity of the fan and air button.

The third button is the most important button, it decided what the other inputs mean. If the button isn’t pressed and other buttons aren’t, the state will continue as it is. If the button isn’t pressed and other buttons are, then the buttons will cause the state to be go ‘backwards’ (backwards meaning that the heat intensity will go down and or the speed will go down). Vice versa if the button is pressed. F, H, And B stand for fan speed, heat intensity and button inversion respectively. The 1 or 0 beside that refers to whether that input is activated or not.

State 0: Off  
State 1 output: Fan on medium and Heat intensity off.  
State 2 output: Fan on medium and Heat intensity on medium.  
State 3 output: Fan on medium and Heat intensity on high.  
State 4 output: Fan on high and Heat intensity off.  
State 5 output: Fan on high and Heat intensity on medium.  
State 6 output: Fan on high and Heat intensity on high.

## State Table for FSM

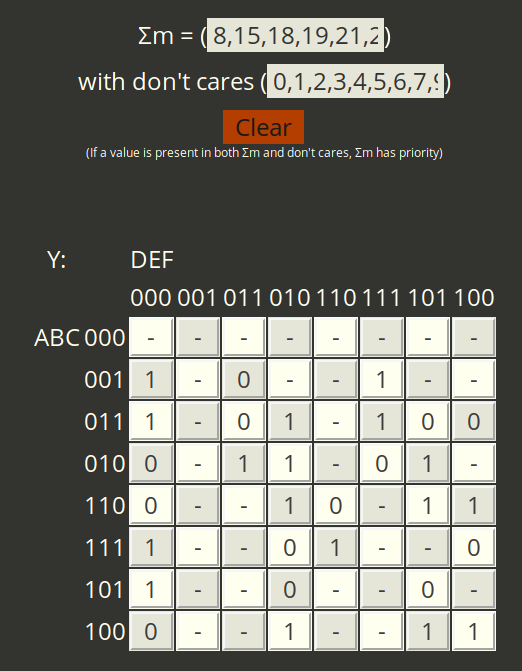
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Current State | Heat | Fan | Button | Next State |
| S0 | 0 | 0 | 0 | S0 |
| S0 | 0 | 1 | 1 | S1 |
| S0 | 1 | 1 | 1 | S2 |
| S1 | 0 | 0 | 0 | S1 |
| S1 | 0 | 1 | 1 | S4 |
| S1 | 1 | 0 | 1 | S2 |
| S1 | 1 | 1 | 1 | S5 |
| S1 | 0 | 1 | 0 | S0 |
| S2 | 0 | 0 | 0 | S2 |
| S2 | 1 | 0 | 0 | S1 |
| S2 | 0 | 1 | 0 | S0 |
| S2 | 1 | 0 | 1 | S3 |
| S2 | 0 | 1 | 1 | S5 |
| S2 | 1 | 1 | 1 | S6 |
| S3 | 0 | 0 | 0 | S3 |
| S3 | 0 | 1 | 0 | S0 |
| S3 | 1 | 0 | 0 | S2 |
| S3 | 1 | 0 | 1 | S6 |
| S4 | 0 | 0 | 0 | S4 |
| S4 | 0 | 1 | 0 | S1 |
| S4 | 1 | 0 | 1 | S5 |
| S5 | 0 | 0 | 0 | S5 |
| S5 | 1 | 0 | 0 | S4 |
| S5 | 0 | 1 | 0 | S2 |
| S5 | 1 | 1 | 0 | S1 |
| S5 | 1 | 0 | 1 | S6 |
| S6 | 0 | 0 | 0 | S6 |
| S6 | 1 | 0 | 0 | S5 |
| S6 | 0 | 1 | 0 | S3 |
| S6 | 1 | 1 | 0 | S2 |

## Next State Logic

Truth table i)

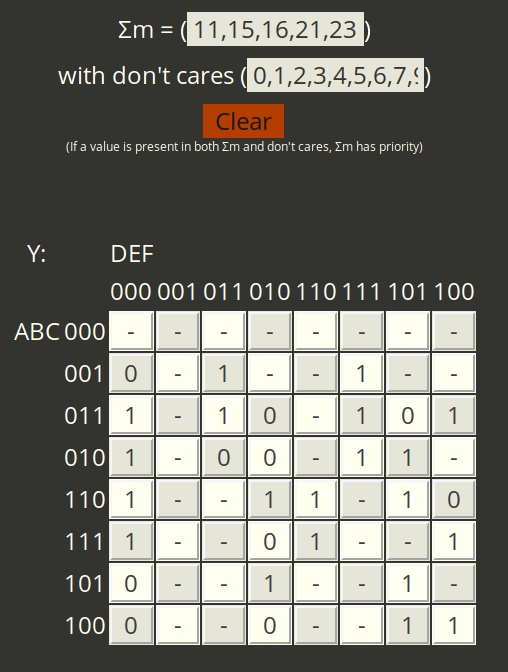
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Current State | | | Heat | Fan | Button | Next State | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

## Karnaugh Maps ii)



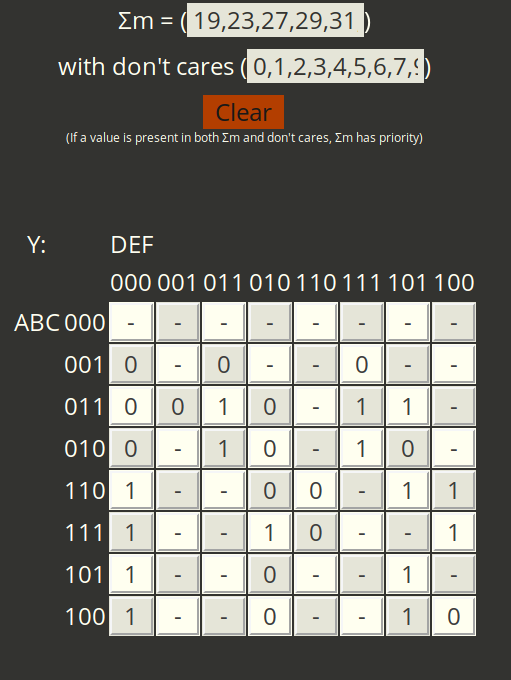
N2

Grouping: C’DE’ + D’C’D + C’D’E + A’EF’ + C’D’F + CD’E’ + CDE



N1

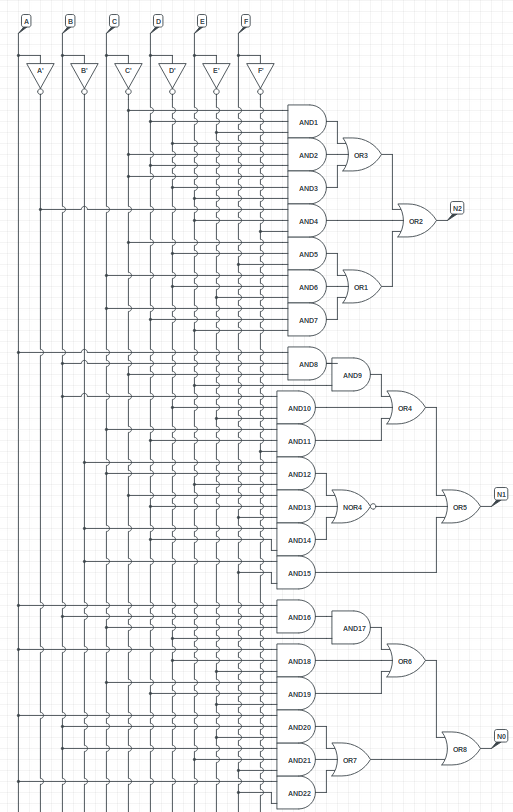
Grouping: ABC’E + BD’E’ + CDF’ + B’CE + C’DF + CEF + B’D + B’F



N0

Y = ABCD’ + AD’E’ + CDE’ + ABE’ +BEF + AF

## Next State Login iii)



## References:

Ref 1: https://en.wikipedia.org/wiki/Finite-state\_machine

Ref 2: <https://er.yuvayana.org/finite-state-machine-history-definition-model-example/>

Ref 3: <https://brilliant.org/wiki/finite-state-machines/>

Ref 4: <https://en.wikipedia.org/wiki/Automata_theory>

Ref 5: <https://medium.com/@mlbors/what-is-a-finite-state-machine-6d8dec727e2c>

Ref 6: <https://www.quora.com/What-is-a-finite-state-machine>

Figure 2:2 - <https://upload.wikimedia.org/wikipedia/commons/thumb/2/2a/CPT-FSM-abcd.svg/326px-CPT-FSM-abcd.svg.png>

Ref 7: <https://nuigalway.blackboard.com/bbcswebdav/pid-1651548-dt-content-rid-12382693_1/courses/1819-CT101/1_CT101_FSM_Design.pdf>

Figure 5: <https://www.tutorialspoint.com/automata_theory/images/state_diagram_of_mealy_machine.jpg>

Figure 6:  
<https://www.tutorialspoint.com/automata_theory/images/moore_machine_state_diagram.jpg>

Ref 8: <https://www.tutorialspoint.com/automata_theory/moore_and_mealy_machines.htm>

Ref 9: https://www.geeksforgeeks.org/difference-between-mealy-machine-and-moore-machine/